

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Applicant(s) : HINSLEY, Christopher, Andrew                      Group Art Unit: TBA  
                      RENOUF, Timothy    Examiner: TBA

International Application No.: PCT/GB00/03172

Serial No : TBA

Filed : February 25, 2002

For : TRANSLATING AND EXECUTING OBJECT-ORIENTED  
COMPUTER PROGRAMS

**PRELIMINARY AMENDMENT**

**COMMISSIONER OF PATENTS**  
**Washington, D.C. 20231**

Sir:

Please amend the above-referenced application as follows:

**IN THE CLAIMS:**

Please amend claim 8, 11, 12, 14, 16, 17, 18, 23, 25 and 26-32 as follows:

8. (Amended) A method as claimed in claim 6 in which the fixup tool returns a value which is used to patch a method which gets or puts the value of a field.

11. (Amended) A method as claimed in claim 9 in which the fixup instructions are transferred, substantially functionally unaltered, by the native translator into the native code; the fixup instructions being replaced with native instructions when the native code is bound on the said real physical processor.

12. (Amended) A method as claimed in claim 1 in which the bytecode is

stack-based, and in which the virtual processor code is register-based.

14. (Amended) A method as claimed in claim 13 including binding the translated tools into a task, and executing the task in native code on the physical processor.

16. (Amended) A method as claimed in claim 1 which includes:

(c) translating the virtual processor code into a different native code which uses instruction set of a second physical processor.

17. (Amended) A method as claimed in claim 13 including executing the different native code on the second physical processor.

18. (Amended) A computer system adapted to carry out a method as claimed in claim 16.

23. (Amended) A distributed computer system as claimed in claim 19 in which the client devices are hand-held games consoles.

25. (Amended) A distributed computer system as claimed in claim 19 in which the server is further arranged to translate the object-oriented computer program from bytecode into virtual processor code.

26. (Amended) A method as claimed in claim 2, including verifying the integrity of the class bytecode, and any external calls.

27. (Amended) A method as claimed in claim 2, in which the class file is a Java class file.

28. (Amended) A method as claimed in claim 2 in which the step of translating the program bytecode into virtual processor code is carried out by a first translator program which is itself written in virtual processor code.

29. (Amended) A Method as claimed as claimed in claim 1, in which the step

of translating the virtual processor code into native code is carried out by a second translator program which is itself written in virtual processor code.

30. (Amended) A computer program for executing a method as claimed in claim 1.

31. (Amended) A data carrier which carries a computer program for executing a method as claimed in claim 1.

32. (Amended) A data stream representative of a computer program for executing a method as claimed in claim 1.

### AUTHORIZATION

If the Commissioner determines that an additional fee is due, or an additional extension of time required, applicant petitions for the extension of time, and authorizes the Commissioner to charge any required fee, or credit any overage, to deposit account 13-4500, Order No. 4362-4002. A duplicate of this sheet is included.

Respectfully submitted,  
MORGAN & FINNEGAN, L.L.P.

By: 

Walter G. Hanchuk  
Reg. No.: 35,179

Date: February 25, 2002

MORGAN & FINNEGAN, L L P  
345 Park Avenue  
New York, NY 10154  
(212)758-4800 / (212)751-6849 (facsimile)

APPENDIX

8. (Amended) A method as claimed in claim 6 [or claim 7] in which the fixup tool returns a value which is used to patch a method which gets or puts the value of a field.

11. (Amended) A method as claimed in claim 9 [or claim 10] in which the fixup instructions are transferred, substantially functionally unaltered, by the native translator into the native code; the fixup instructions being replaced with native instructions when the native code is bound on the said real physical processor.

12. (Amended) A method as claimed in [any one of claims 1 to 11] in claim 1 which the bytecode is stack-based, and in which the virtual processor code is register-based.

14. (Amended) A method as claimed in claim 13 [when dependent upon claim 2] including binding the translated tools into a task, and executing the task in native code on the physical processor.

16. (Amended) A method as claimed in [any one of the claims 1 to 12] claim 1 which includes:

(c) translating the virtual processor code into a different native code which uses instruction set of a second physical processor.

17. (Amended) A method as claimed in claim 13 [or claim 14] including executing the different native code on the second physical processor.

18. (Amended) A computer system adapted to carry out a method as claimed in claim 16 [or claim 17].

23. (Amended) A distributed computer system as claimed in claim 19 [or claim 20] in which the client devices are hand-held games consoles.

25. (Amended) A distributed computer system as claimed in [any one of

claims 19 to 24] claim 19 in which the server is further arranged to translate the object-oriented computer program from bytecode into virtual processor code.

26. (Amended) A method as claimed in [any one of claims 2 to 11, or claim 12 when dependent upon claim 2] claim 2, including verifying the integrity of the class bytecode, and any external calls.

27. (Amended) A method as claimed in [any one of claims 2 to 11, or claim 12 when dependent upon claim 2] claim 2, in which the class file is a Java class file.

28. (Amended) A method as claimed in [any one of claims 1 to 12, 26 or 27] claim 2 in which the step of translating the program bytecode into virtual processor code is carried out by a first translator program which is itself written in virtual processor code.

29. (Amended) A Method as claimed as claimed in [any one claims 1 to 12, 26, 27 or 28] claim 1, in which the step of translating the virtual processor code into native code is carried out by a second translator program which is itself written in virtual processor code.

30. (Amended) A computer program for executing a method as claimed in [any one of claims 1 to 12 or 26 to 29] claim 1.

31. (Amended) A data carrier which carries a computer program for executing a method as claimed in [any one of claims 1 to 12 or 26 to 29] claim 1.

32. (Amended) A data stream representative of a computer program for executing a method as claimed in [any one of claims 1 to 12 or 26 to 29] claim 1.